Report on StyleGAN-NADA Re-implementation

Introduction

This research explores and re-implements the StyleGAN-NADA image editing method (Fig. 1). The goal is to fine-tune a generative model to create images from a chosen domain, including abstract or non-realistic ones. A key feature of this method is that it doesn't require images from the target domain for training; instead, it uses a textual description of the goal, which eliminates the need for collecting and labeling additional data.

The foundation of our work is **StyleGAN2**, an architecture composed of a Mapping Network (which transforms random noise into a style-based latent vector W) and a Synthesis Network (which uses these W vectors to generate an image) (Fig. 2). We learned to modify the Synthesis Network of the generator while preserving the structure and properties of the Mapping Network and the overall latent space. This approach, after relatively quick training, allows for the generation of an unlimited number of diverse images within the selected domain.

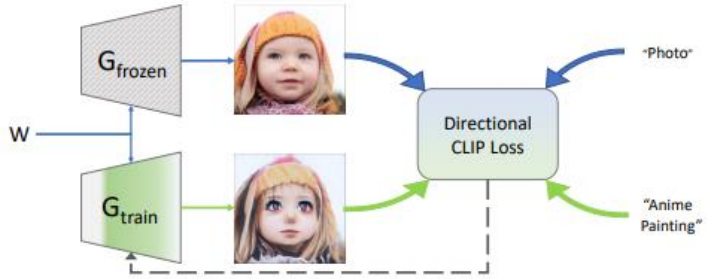

Figure 1. StyleGAN-NADA.

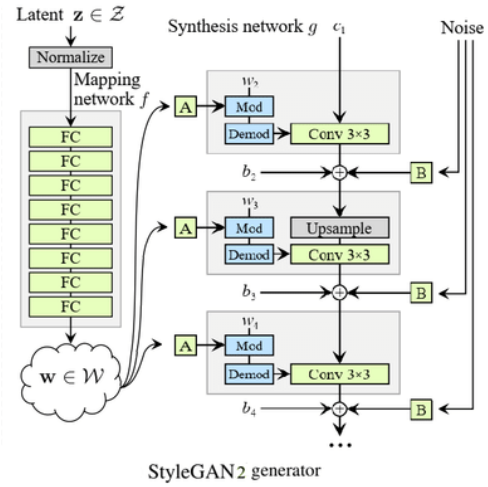

Figure 2. StyleGAN2.

The **CLIP (Contrastive Language-Image Pre-training)** model was used to create the textual representation (embedding). CLIP is a multimodal model that "understands" the relationship between images and text by mapping them into a unified semantic space. CLIP can assess how an image's style should change based on a text prompt, providing vector representations for both images and text that allow for measuring their semantic similarity (Fig. 3).

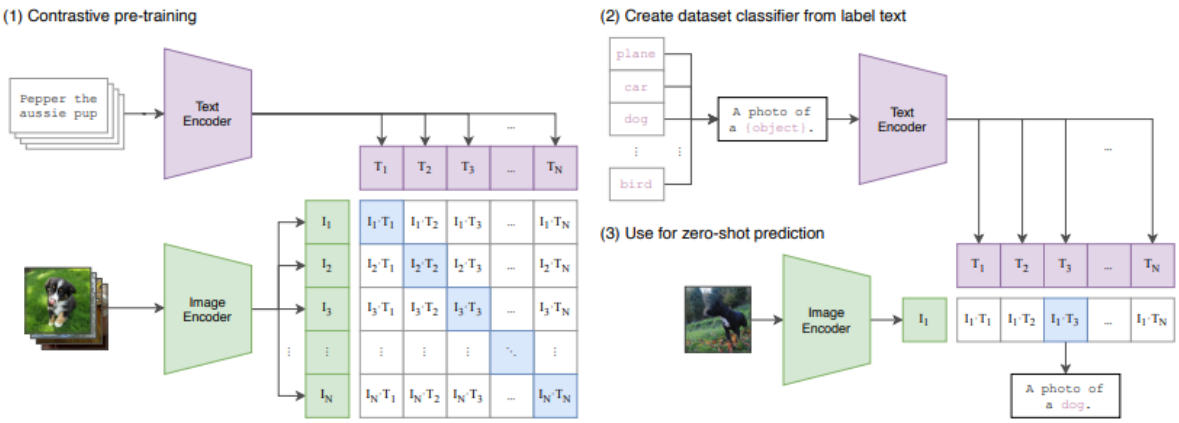

Figure 3. CLIP (Contrastive Language-Image Pre-training).

Operating principle

Training is performed using two instances of the generator, which are initially identical copies of a pre-trained StyleGAN2.

1. The first generator (`model_frozen`) is frozen and used to generate images from the source domain. It serves as an "anchor" or a reference point.
2. The second generator (`model_train`) is trained to generate images for the target domain.

The core idea of the training is to modify the "direction" of the `model_train`'s generation to align with the "direction" in the CLIP space defined by the text prompts. This means that `model_train` adapts to move the image modification vector along the vector direction between the source text (`source_text`) and the target text (`target_text`) in CLIP space.

For this purpose, a **Directional CLIP Loss** function is used. This loss seeks to align:

- The change vector in the image CLIP space (obtained from `CLIP(generated_img_train) - CLIP(generated_img_frozen)`).
- With the change vector in the text CLIP space (obtained from `CLIP(target_text) - CLIP(source_text)`).

The directional loss function uses the cosine similarity between these two directional vectors and is defined as:

$$L_{CLIP\_direction} = 1 - cosine\_similarity(CLIP(I_{train}) - CLIP(I_{frozen}), CLIP(T_{target}) - CLIP(T_{source})) \quad (1)$$

Where:

- I_{train} — the image generated by the trainable generator.
- I_{frozen} — the image generated by the frozen generator.
- T_{target} — the target domain text prompt.
- T_{source} — the source domain text prompt.

Loss function

The overall loss function (`Total Loss`) is the sum of two main components:

$$L_{total} = \lambda_{CLIP} \cdot L_{CLIP\_direction} + \lambda_{L2} \cdot L_2 \quad (2)$$

1. $L_{CLIP\_direction}$ (CLIP Directional Loss): This is the main loss that directs the generator's adaptation in the CLIP space. It compels the generated images to move in the direction specified by the text prompts.
2. L_2 (L2 Reconstruction Loss): This reconstruction loss measures the difference between the generated images I_{train} and I_{frozen}. It helps to preserve the overall structure and details of the original images, preventing the generator from deviating too much from realism or losing identity during the transformation.

Dynamic adjustment of loss coefficients

To address the challenge of selecting the coefficients λ_{CLIP} and λ_{L2} we used a dynamic method involving a separate optimizer, `optimizer_lambda`, which updates these coefficients using a gradient descent approach.

Advantages of this approach:

- Automatic Tuning: Instead of the laborious manual tuning of loss weights (`lambda_clip`, `lambda_l2`), the model automatically finds their optimal values based on the overall loss function.
- Dynamic Adaptation: The loss weights can change during training, adapting to the model's current state and gradients. For example, if $L2$ becomes too large (the model deviates significantly from the original space), while $L_{CLIP\_direction}$ is already small enough, the optimizer can automatically increase λ_{L2} and/or decrease λ_{CLIP}, to focus on preserving structure. This ensures more balanced training.
- Reduced Manual Work: It lessens the need for extensive hyperparameter searching, which significantly speeds up the experimentation cycle.

Selection of layers for unfreezing in the trainable generator

It is known that not all generator layers are equally important for introducing stylistic or domain-specific changes. Some layers are responsible for low-level features (overall structure), while others are for high-level details (textures, color, "style").

To find the most sensitive layers for unfreezing (i.e., those that will participate in training), we tested several methods:

1. Unfreezing only late Synthesis Network Layers (Experiment 1, `freeze_layers`):

Approach: Only the late layers of the Synthesis Network were unfrozen. These layers are typically responsible for generating finer details and textures at high resolution. The unfreezing was done once, before the training loop.

Result: The training outcome was unsatisfactory, which likely indicates that these layers have insufficient influence for proper adaptation to new, more global domains (Fig. 4).

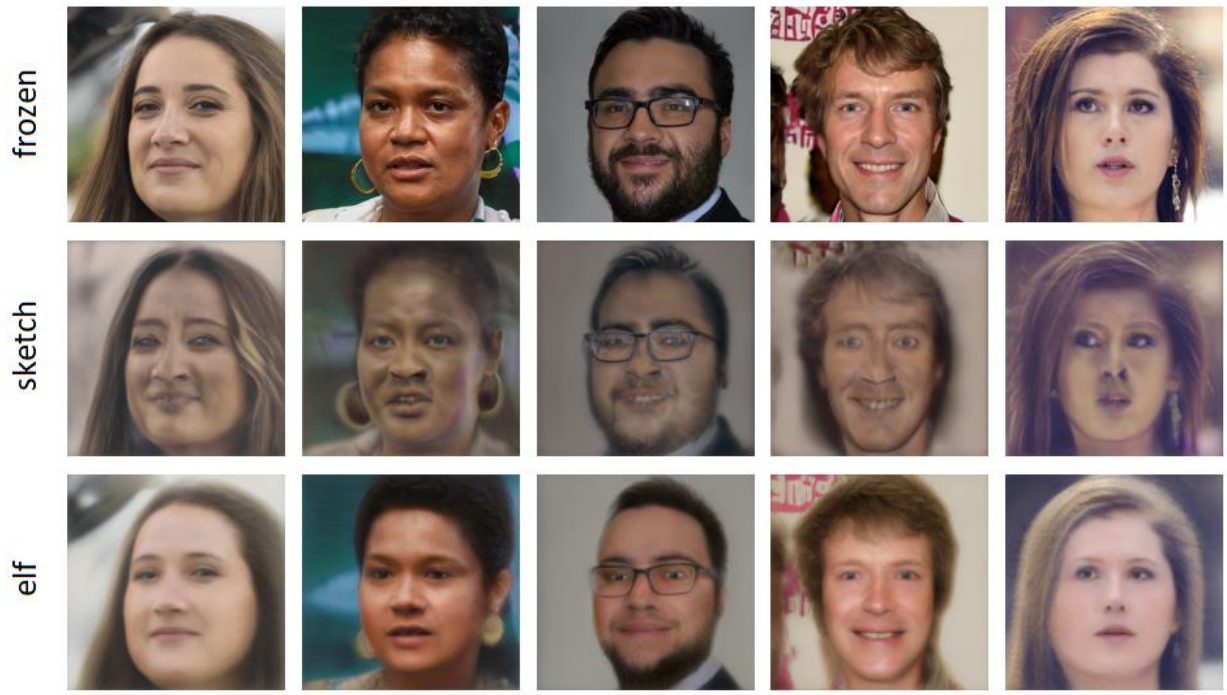

Figure 4. freeze_layers.

2. Adaptive layer freezing method (Original StyleGAN-NADA approach, Method 2, `freeze_layers_adaptive`):

Approach: We used the adaptive layer freezing method as proposed in the original paper. This means that at each training iteration, only the most relevant layers of the generator are selected for updating, while the rest are frozen. This method automatically tries to identify the most important layers to modify based on their influence on the W-space (latent codes).

Details of `freeze_layers_adaptive`:

- W+ vector creation: A random Z vector is generated, transformed into a W vector using the frozen generator, and then "broadcast" into a W+ form to have a separate `w` for each generator layer.
- W-vector mini-optimization: Several steps of gradient descent are performed to find a modified W vector that better aligns with the target text in CLIP space. The generator itself is not trained; only the W vector is optimized.
- Layer Importance Assessment: The magnitude of change for each W+ vector component corresponding to a specific generator layer is measured during this mini-optimization (by calculating the absolute difference `torch.abs(latent_tensor - latent_w_plus)`). It is assumed that the layers requiring the largest W-vector changes to achieve the goal are the most "important" for the current stylization task.
- Selection of k most important layers: The `k` layers with the largest changes are selected.
- Generator Freeze/Unfreeze: All parameters of `model_train` are first frozen by default. Then, only the parameters belonging to the selected `k` layers are unfrozen.

Result: The outcome was good. This approach enhances training stability and efficiency by focusing the learning on the most influential parts of the model (Fig. 5).

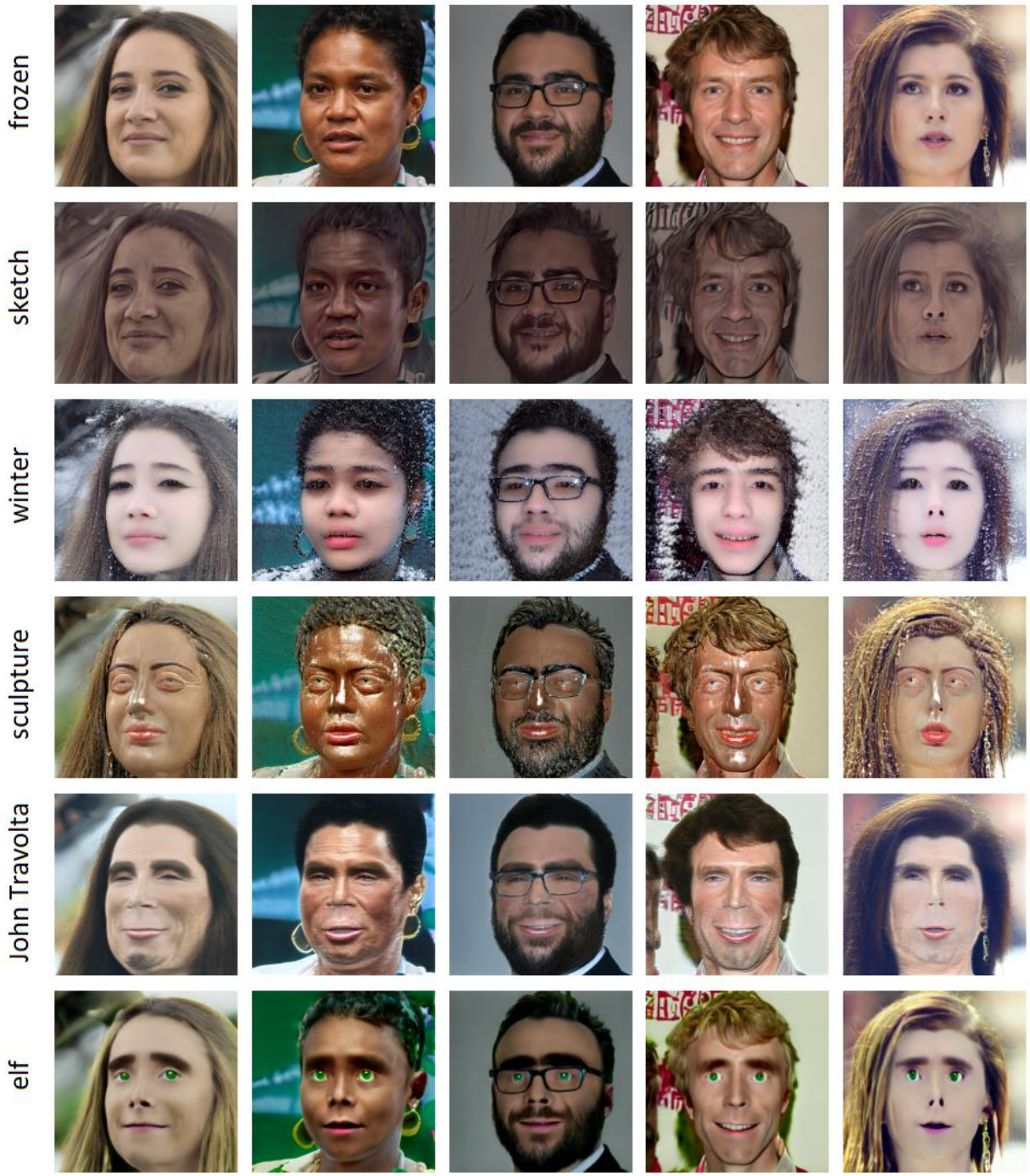

Figure 5. freeze_layers_adaptive.

3. Adaptive method with partial layer unfreezing (Experiment 3, `freeze_layers_adaptive_fine_grained`):

Approach: We tested a similar adaptive method where important layers are automatically identified. The difference is that the selected layers are not completely unfrozen: the `weight` and `bias` parameters remain frozen, while `modulation` and `noise` parameters are unfrozen.

Reason for choosing `modulation` and `noise`: In StyleGAN, `modulation` (adaptive instance normalization) is responsible for applying styles controlled by the `w`-vector, and `noise` adds stochastic details. Training only these parameters allows for changes in style and details while preserving the core "filters" (convolutional weights) responsible for structural features. The function

thus "flexibly" unfreezes only the most important parts of the model directly responsible for applying style.

Result: The outcome was unsatisfactory. This may indicate that to achieve the desired domain changes, especially if they affect not just style/texture but also more global shapes, the base weights of the convolutional layers also need to be unfrozen (Fig. 6).

Figure 6. freeze_layers_adaptive_fine_grained.

4. Adaptive method with selection of most sensitive parameters (Experiment 4, `freeze_layers_adaptive_fine_tune`):

Approach: We tested a more granular adaptive unfreezing method by identifying the most significantly changing parameters rather than entire layers.

Details: At each epoch, the function calculates the change in generator parameters during a minimal optimization (an internal loop training a copy of the training generator) to understand which specific weights are truly participating in directing the image towards the given text. Then, the `k` weights with the largest changes are unfrozen for further training of `model_train`.

Result: This method proved to be too "granular," not giving the generator enough flexibility for broader domain changes (Fig. 7). Moreover, the results were unpredictable, as unfreezing only specific parameters within a layer was insufficient. This meant that during training, `model_train` did not make the necessary shift in the visual space to align with the direction in the text space.

Figure 7. freeze_layers_adaptive_fine_tune.

Selection of Source text

Since CLIP was trained on generalized internet images and might not interpret local visual changes as we expect (e.g., due to bias in the training data), we tested a method of dynamically updating the source text (`source_text`) at each epoch. This update is based on the CLIP description of the image generated by the frozen generator (Fig. 8).

Idea: CLIP "on the fly" determines "what the image sees" that was generated by `model_frozen`, and this becomes the current starting point (`source_text`).

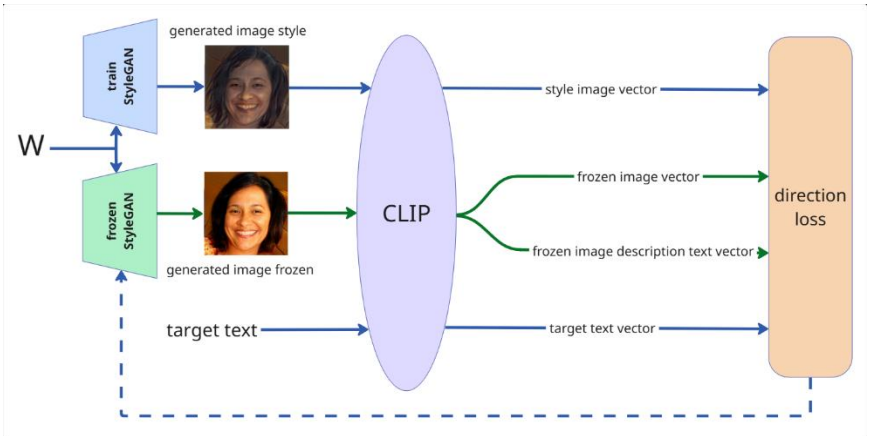

Figure 8. Selecting the source text with CLIP.

In other words, we want the directions between the image and text vectors to not be too random or meaningless. We aim for CLIP to "understand" where we want to go and for the difference between `target_text` and `source_text` to be adequately reflective of the current state of `model_frozen`. It is hypothesized that the visual shift made by the model will align better directionally with the textual shift if the `source_text` is a dynamically generated description.

For comparison, this method was tested with the adaptive layer freezing method (`freeze_layers_adaptive`).

Result: The visual results obtained did not differ from using the adaptive layer freezing method without dynamic source text updates. The use of this method did not add any noticeable benefits (Fig 9).

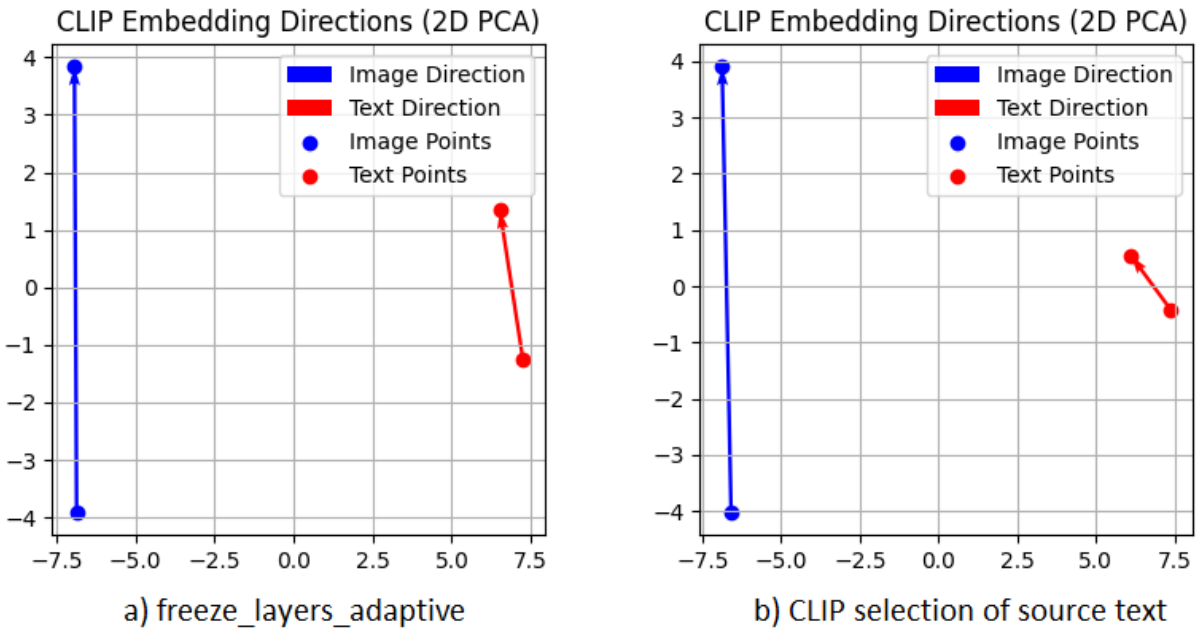

Figure 9. Visualization of CLIP Directions.

Conclusions

We have successfully reproduced and reimplemented the StyleGAN-NADA workflow, learning to fine-tune a generator to create images from various target domains based solely on text prompts.

Key advantages of StyleGAN-NADA:

- **No need for training images for the target domain:** a text prompt is sufficient. This significantly reduces the cost of data collection and labeling.
- **Flexibility of transformations:** The method allows for transforming images into a wide range of domains, including stylistic changes, shape modifications, and even the generation of fantastical or abstract images.
- **High training speed:** Thanks to fine-tuning a small number of layers (rather than training from scratch) and the efficient use of CLIP, the adaptation process is significantly faster than traditional GAN training.
- **Preservation of latent space properties:** The method maintains the structure of the StyleGAN latent space, which allows for further editing of the generated images.
- **Preservation of generation quality:** We confirmed that adaptively freezing/unfreezing generator layers helps preserve the base generation quality and structure of the original image, as the unmodified layers help maintain realism.

Drawbacks and limitations:

- **Pre-trained generator limitations:** The modifications the method can apply are largely constrained by the domain of the pre-trained StyleGAN2 generator. If the target domain is too different from the source, more profound changes or pre-training on a more diverse dataset may be required.
- **Appearance of artifacts:** With very strong image changes or when adapting to complex/unrealistic domains, artifacts such as distortions or non-realistic details may appear.

- **Variety:** A separate generator training session is required for each style of generated images.

Peculiarities of using CLIP:

We successfully used CLIP for image editing in the direction of a text prompt. This is a flexible and universal approach that allowed us to train the generator using only text prompts. However, CLIP has several limitations:

1. **Limited perception by training data:** CLIP is limited by the data it was trained on. This can lead to "hallucinations" or inaccurate interpretations for highly specific or rare visual concepts not present in its training set.
2. **Limited semantic control accuracy:** CLIP doesn't always precisely understand subtle visual differences (e.g., "shiny hair" versus "wet hair" might be interpreted similarly if these nuances were not clearly represented in its training data).
3. **Text meaning generalization:** CLIP often generalizes the meaning of text, which can lead to unexpected or undesired results during editing if the user expects a very specific transformation.
4. **Lack of pixel-level feedback:** Unlike a perceptual loss (e.g., VGG-loss), CLIP does not compare pixels directly but works with global image and text embeddings. This means that the results may not always perfectly align with the user's visual expectations, even if CLIP "thinks" the result is semantically correct.

Tested Training Configurations:

The most successful configurations we tested were:

- Automatic selection of loss function coefficients (λ_{CLIP} and λ_{L2}) using a separate optimizer.
- Adaptive freezing and unfreezing of generator layers (the approach described in point 2, "Adaptive Layer Freezing Method").

Real image editing implementation:

In combination with an image inversion method (pSp/e4e - encoder4editing), StyleGAN-NADA was used to edit real images (Fig. 10). The process involves inverting a real image into the StyleGAN latent space and then applying the adapted generator to transfer that image to a new domain while preserving its identity.

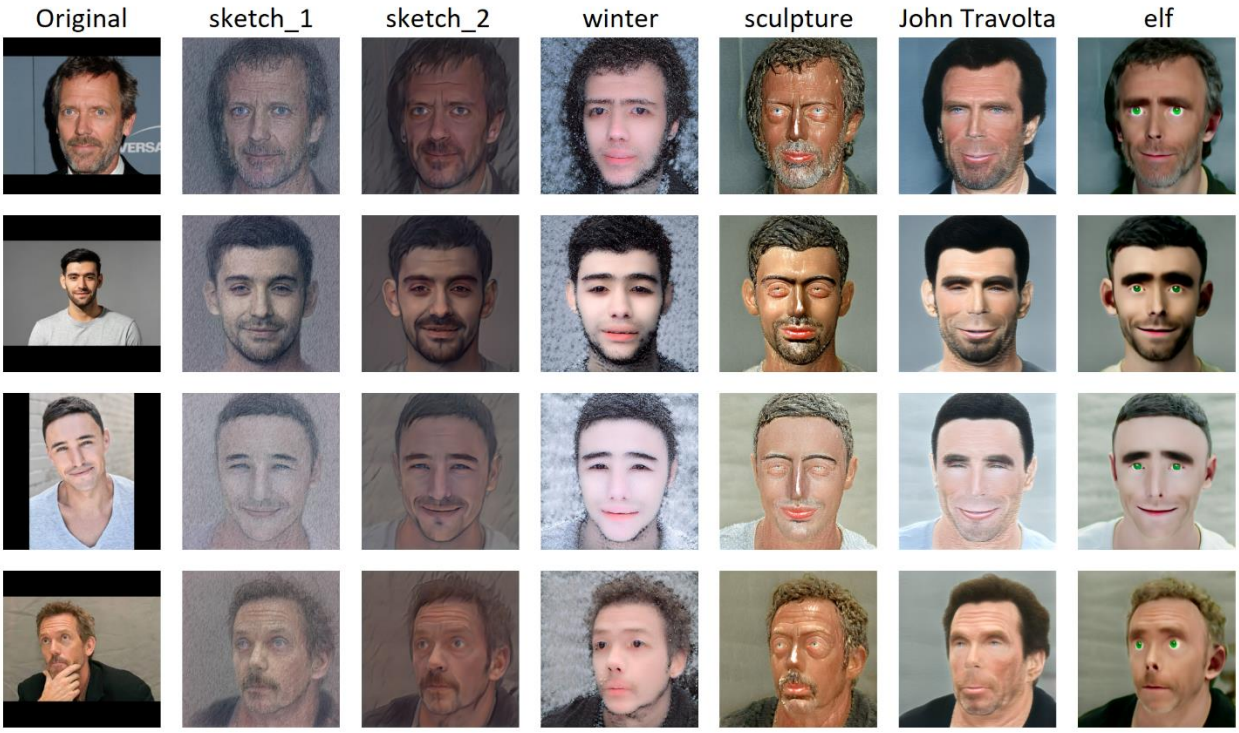

Figure 10. Editing real images.